

How do you retrieve a Web page?

* When you enter a URL in your browser, your computer typically performs the following actions.

- 1) It uses DNS to look up the IP address of the Server.
- 2) It establishes a connection to that IP address and requests the specified file.
- 3) The web browser software, retrieves the file from the server's local file system and sends its content back to your browser.

Server Side pages

Server-Side pages usually refer to programs that are written using various Web programming languages or frameworks.

Eg:- PHP, Ruby, Rails

These programs are executed by the Web Server according to the Client requests, and output is sent back to the Client as a response.

PHP - Hypertext preprocessor

- * A popular Server-Side Scripting language.
- * used for creating dynamic Web pages.
- * PHP can act as an interface with other Web services such as databases, email servers, and other APIs.
- * Also handles user authentication and processes form data.
- * It can be embedded directly into HTML code.

PHP based web servers

- * Apache HTTP Server * Nginx
- * Lighttpd * Caddy
- * Microsoft IIS

- * Each PHP statement is an instruction that must be followed by the Web server.

```
< ?php
    echo "Hello World...";
```

Opening PHP tags

? ↑
PHP output function

String

↑ End of statement

closing PHP tags

echo

print

- * Can output multiple expressions Only output one.
Separated by commas, i.e. expression at the

echo "Hello", "World";

print "Hello World";

echo fast because it does not return value

Print returns a value of 1

Variables

- * Variable names
 - * can contain letters & numbers and underscores.
 - * must start with a letter or underscore.
 - * Cannot start with a number.
 - * Variable names can be ^{contain} multiple words and the words in the variable can be separated by - (underscore).
 - * Variable names always begin with \$, on both declaration and usage.
 - * a single variable may contain any type of data.

Data types

- * Integer
- * Object
- * Float / Double
- * Null
- * String
- * Boolean
- * Array
- * Array

- * Single line comments - // or ##
- * multi-line /* */

array

```
$array1 = array(1, 2, 3); { echo array[0];  
$array1 = [1, 2, 3]; }
```

```
$fruits = array ("apple", "banana", "orange");  
$names = ["name" => "John", "age" => 30, "gender" => "Male"]
```

Date: _____

No: _____

Add element

$\$array[0] = "C"$;

$\$names["City"] = "Battaramulla";$

HTML forms

* Essential Component of Web Development

GET POST method

- * These methods used to pass data between a Client and a server-side PHP script
- * Both GET & POST create an array

GET

POST

- * GET method sends data to the Server as part of the URL
- * POST method sends data to the Server in the request body.

* This method used for retrieving data from the Server

* POST method used for modifying data on the Server.

* It is suitable for transmitting small amounts of non-sensitive data.

* This is suitable for transmitting large amount of sensitive data.

Superglobal Variables

- * predefined Variables that are available in all scopes throughout a PHP Script.

9 Superglobal variables

\$-SERVER:

- * Contains information about the current script's execution environment such as,
 - * Current request method
 - * headers
 - * Server Information

\$-GET :

- * Contains data that is passed to the script through the URL

\$-POST :

- * Contains data that is passed to the script through the HTTP POST method

\$-FILES :

- * Contains information about uploaded files.
Such as
 - * name
 - * size
 - * type

\$-COOKIE

- * Contains data that is stored in cookies

\$-SESSION

- * Contains data that is stored in a session

\$request

- * Contains data, ^{that are} passed to the script through GET method, post method and cookies.

No: _____ Date: _____

\$-ENV: Contains information about system environment
such as
* environment variables

\$GLOBALS: used to access global variables from within a function or method

Validating user inputs

- * required fields - ISSET(), empty()
- * Data Type - IS_NUMERIC(), IS_STRING()
- * Length limits - strlen()
- * Regular Expression - Check if the input matches a specific pattern or format.
- * Range limits - <=, >=, <, >

filter_var()

* used to validate and sanitize different types of user input using various flags.

Ex:- filter_var(\$email, FILTER_VALIDATE_EMAIL)

* validating user input means,

This is important step in processing form data in PHP, as it helps to ensure that the user inputted data is in the expected format.

Sanitizing user inputs:

- * Sanitizing is the process of removing any potentially harmful or unwanted data from user inputs.
- * Sanitizing typically involves removing any HTML or script tags from user input.
- * Several PHP functions that can be used for sanitizing

* trim()

* htmlspecialchars()

* stripslashes()

* htmlspecialchars()

* strip_tags()

Query String

- * It used to pass information from one page to another through the URL
- * They allow you to append key-value pairs to a URL, which can be accessed and processed by the receiving page.
- * \$_GET superglobal variable is used to retrieve query string parameters passed in the URL

http://www.example.com/page.php?key1=value&key2=value2

No: _____

Date: _____

Dynamic Content may load
slower, especially if it
involves complex operations
or database interactions.

Ex:- E-commerce websites
Social media platforms
Content management
systems.

Object-Oriented programming

Classes

- * Classes are the fundamental building blocks of OOP
- * A class can be defined as a template/blueprint that describes the properties (attributes) and methods of an object.

Objects

- * Object is an instance of a class.
- * Real world entity with states & behaviours.
- * States means entity attributes
- * behaviours means, object do multiple activities they called behaviours.

Encapsulation

- * Wrapping up attributes & behaviours of the object together into one unit.
- * It hides the internal implementation details.
- * exposes only the necessary interfaces to interact with the object.
- * Encapsulation ensures data security and code organization.

Inheritance

- * The ability of creating new classes based on existing classes.
- * (Parent class) Super class provide their attributes and behaviours to subclass / child class.
- * Inheritance promotes code reuse and allow the creation of specialized classes.

Polymorphism

- * Ability of an object to exists in many forms.
- * This promotes code modularity and simplifies adding new functionality.
- * This allows flexibility in writing code that can work with objects of different types.

Abstraction

- * The process of hiding the implementation details and showing only essential features (functionality) to the user.
- * Abstract classes and interfaces are used to create common behaviours and characteristics shared by multiple objects.
- * Abstraction allows programmers to work with simplified models of complex systems.

Properties.

- * also known as attributes or ~~beta~~ variables.
- * properties that will hold the data for each object of the class.

Methods

- * also known as functions.
- * methods that will perform actions or provide behavior for the objects of the class.

Constructor

- * It is a special method that is automatically called when an object of a class is created.
- * It is used to initialize the object's state and perform any necessary setup operations.

Destructor

- * It is automatically called when an object is no longer referenced or when the script execution ends.
- * It is used to perform cleanup tasks, such as
 - * releasing resources (closing files, database connections)
 - * freeing memory

No. _____

public - It can be accessed from anywhere.

protected - It can only be accessed within the class itself and its subclasses.

private - It can only be accessed within the class itself.

Interfaces in PHP

* A set of rules that classes must follow or a set of methods that a class must implement.

* A blueprint for the methods that class should provide, without specifying the implementation details.

* Interface are useful for,

* achieving abstraction

* defining common behaviour.

* promoting code reusability.

* Achieve polymorphism

* provide clear guidelines for implementation.

Abstract classes.

* Abstract classes serve as blueprints for other classes.

* It is designed to be extended by other classes, which can provide implementations for abstract methods and inherit its properties and non-abstract methods.

abstract classes are used to,

- * define common behavior and characteristics for multiple related classes.

- * abstract classes cannot be instantiated.
(cannot create an object of an abstract class)
- * can have both abstract and non-abstract methods.
- * can provide default implementations for non-abstract methods.
- * useful for sharing code and implementing common functionality among related classes.
- * A class can only extend one abstract class
- * A class can implement multiple interfaces.

Polymorphism in PHP

- * the ability of an object to exists in many forms.
 - * can be achieved in two ways
- 1) Method overriding - A child class can provide its own implementation of a method defined in the parent class.
 - 2) Interface implementation.
- * PHP won't support method ~~alias~~ overloading.

requireInclude

- * If the file cannot be found or there is an error during the inclusion process, it halts script execution.
- * If the file cannot be found or there is an error during the inclusion process, it generates a warning message and continues script execution.
- * Used for essential files or dependencies that are required for the script to function correctly.
- * used for non-essential files or dependencies that can be missing without causing critical issues.

~~# require_once and include_once~~

~~# Ensure that a file is included only once~~

use of Namespaces.

- * Similar to packaging concept in Java.
- * Namespaces are particularly useful in larger PHP projects or when working with third-party libraries.
- * Use namespace keyword to define a namespace.
- * You can use the use keyword to import a namespace or specific elements from namespace.
- * A way to Organize your code, to avoid having conflicts and improve code readability and maintainability.
- * If you have multiple classes under one namespace, you should add one by one using use keyword.

No: _____ Date: / /

Dependency Management in PHP

- * The process of managing external libraries, packages and dependencies that your PHP project needs.
- * It involves handling the installation, versioning, and updating of these dependencies to ensure smooth integration and functionality with your project.
- * You can use Composer as a tool to manage your dependencies.